

Why Platform Engineering?



The state of the industry

65% of executives believe improving DevEx is one of the top goals for 2025

58% of digital organizations have either deployed or deploying a developer portal by late 2023

88% of tech executives believe PE is critical in achieving their software engineering goals

Dev Portals + Platform Engineering is what organizations do. By 2026, 75% of the organizations will have that combination, up from 45% in 2024

57% of tech executives believe the most important thing in PE is Platform-As-A-Product

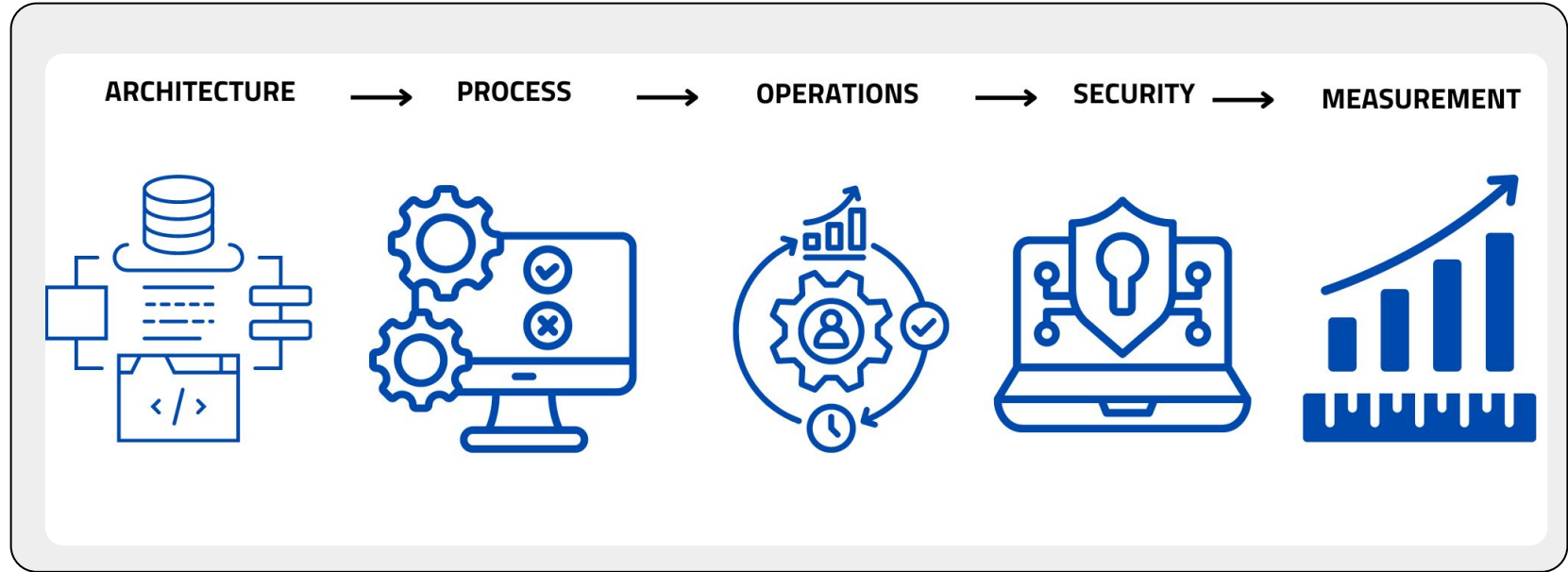
The evolution of Platform Engineering

- Challenges and past trends that led to today's Platform Engineering
- Evolving PE in organizations
- Platform Engineering: Career Paths and Growth Opportunities
- Making it all stick
- The current and future trends in Platform Engineering
- How this book will help you?

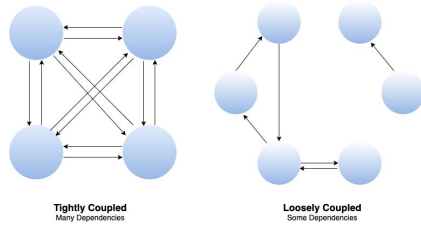
Challenges & Trends

“The story of how the platform engineering evolved”

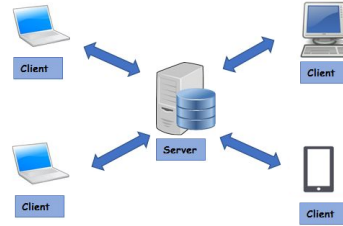
Software (Product) Development Evolution



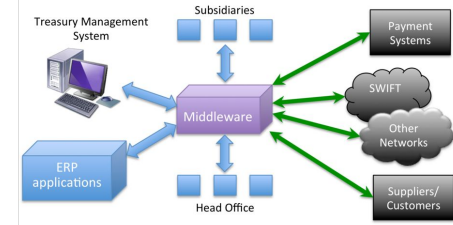
Architectural Milestones



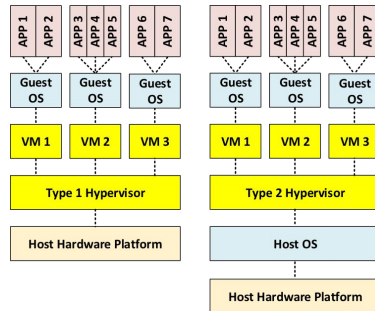
LOOSE COUPLING



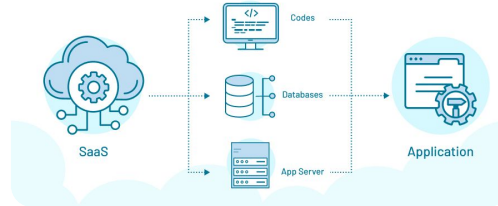
CLIENT / SERVER



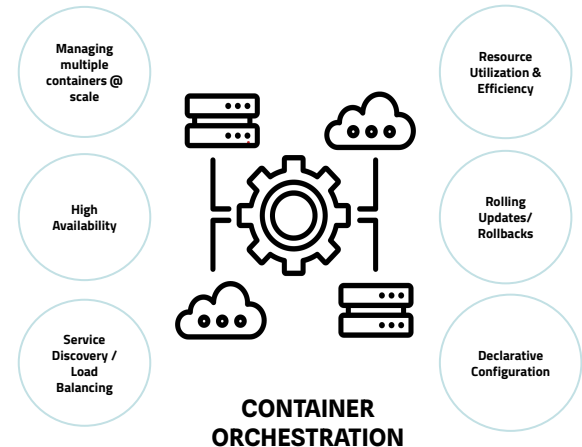
MIDDLEWARE



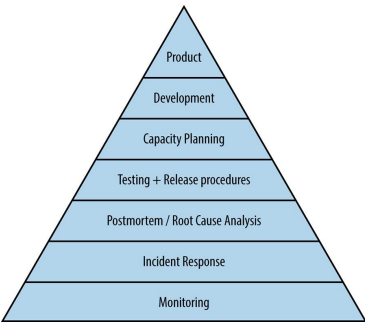
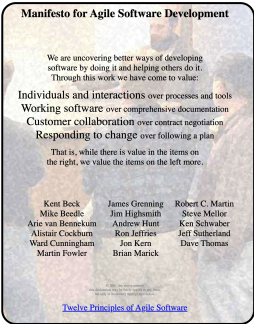
VIRTUALIZATION



SaaS

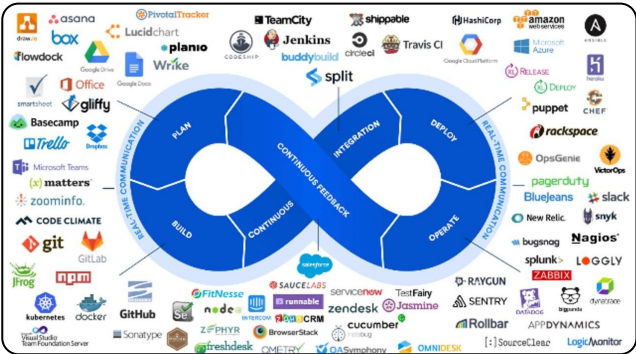


Process Milestones

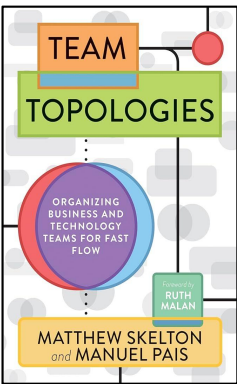
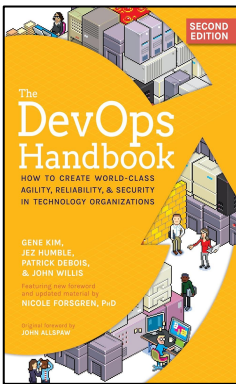
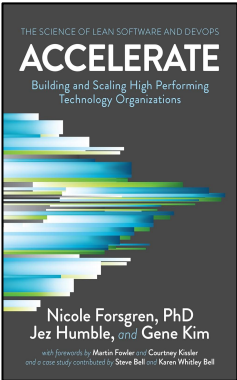
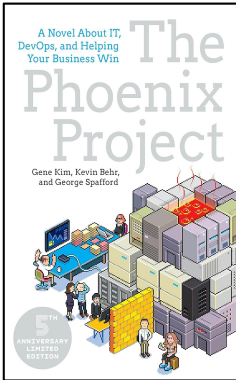


AGILE MANIFESTO

SRE



TOOL PROLIFERATION



SEMINAL WORKS

Operational Milestones



**SOFTWARE
BENCHMARKING**



PORTABILITY



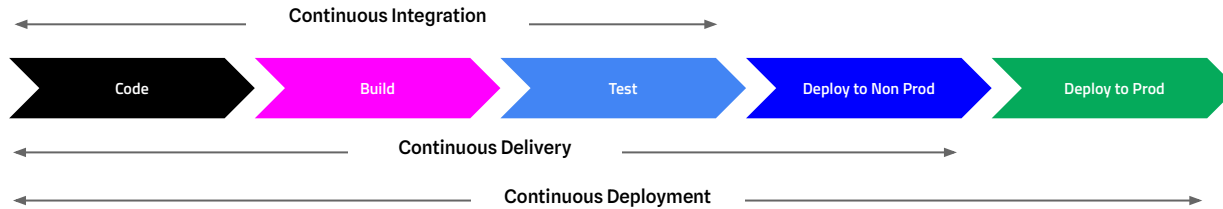
**MEDIA BASED
DELIVERY**



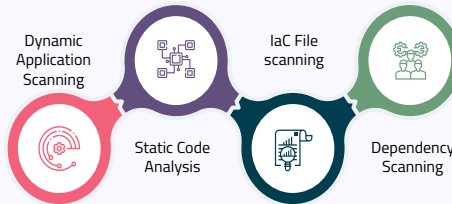
**INSTALLED
SOFTWARE**



INTEGRATING SOFTWARE



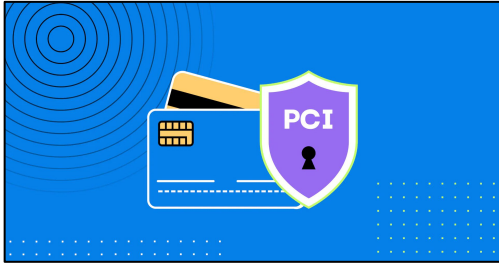
DevSecOps



Continuous Compliance



Security Milestones

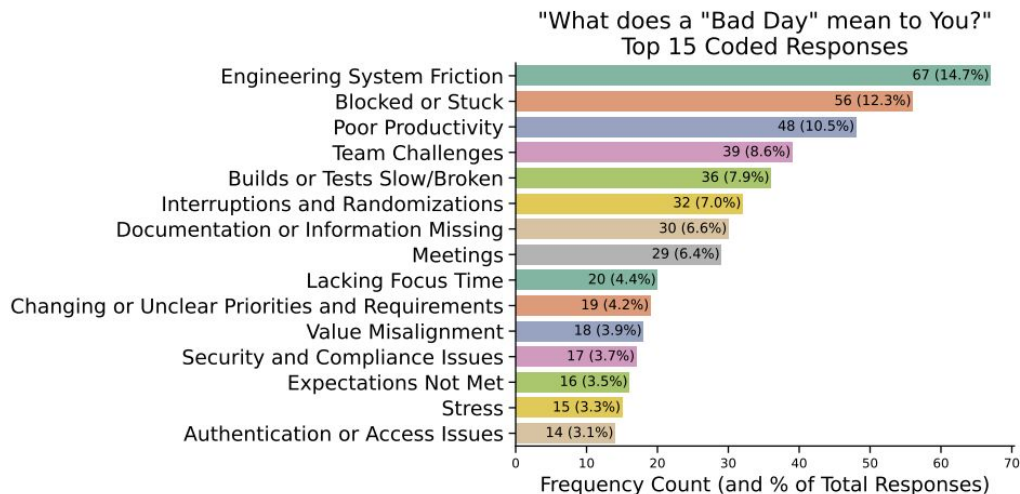


Now, can you start measuring the experience?



In over 50% of organizations, software developers only spent 30-40% of their time on feature development

What causes bad days for developers?

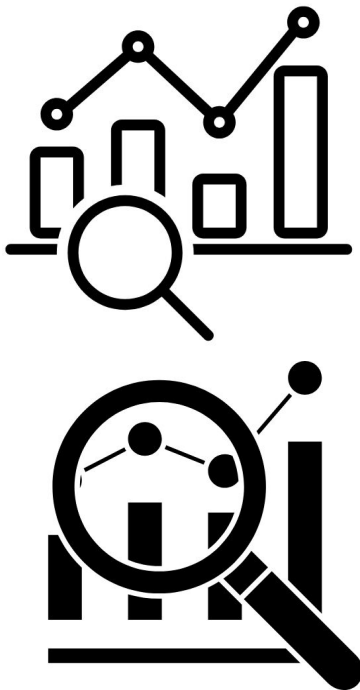


Microsoft research Oct '24 - <https://arxiv.org/pdf/2410.18379>

Focus on DevEx

Qualitative Metrics

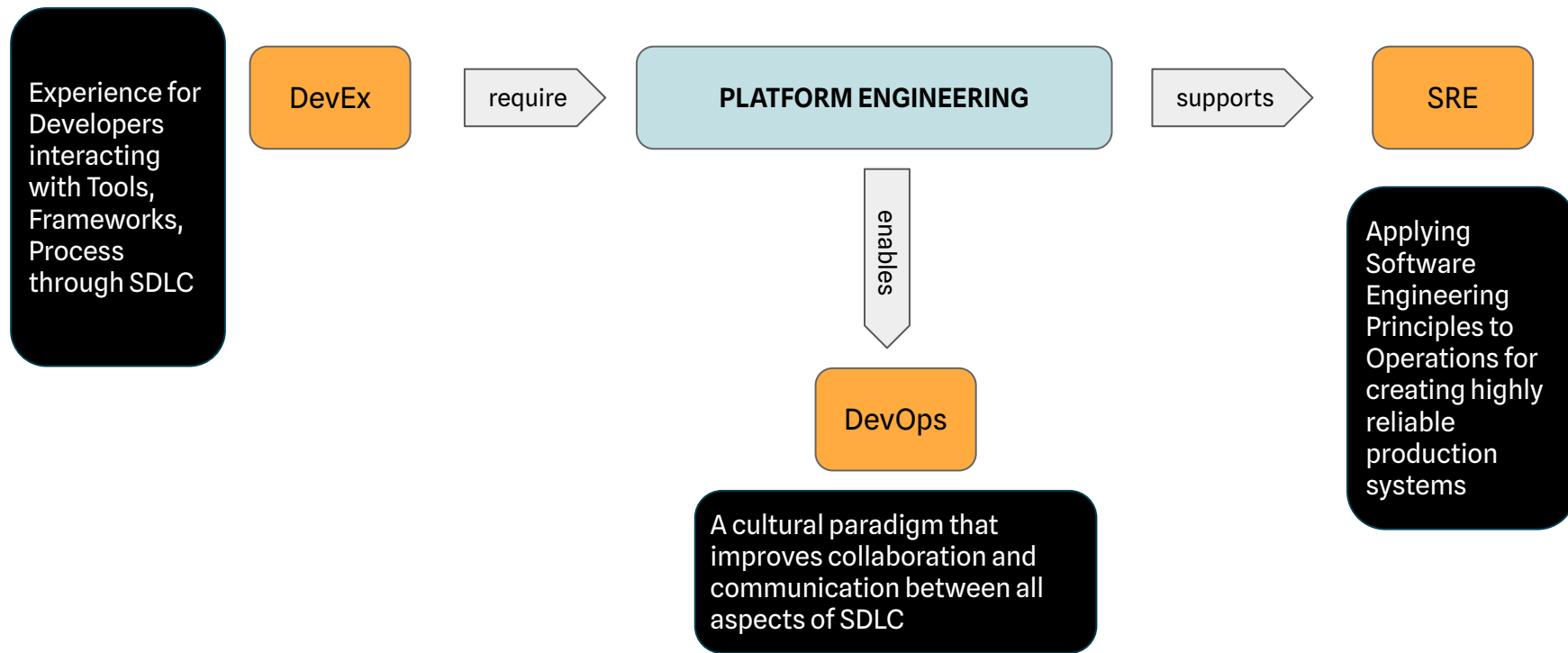
- Developer Surveys
- Empathy Interviews
- Focus Groups
- Developer Forums
- Sentiment Analysis



Quantitative Metrics

- Platform Value Metrics
- DORA / SPACE metrics
- Leading metrics (Code Quality, PR,...)
- NPS
- Operational Metrics

Platforms to the rescue



Platform Engineering

Economies of Scale | Abstracting out complexity

Platform Engineering enables **DevOps** while supporting **SRE**, including **Observability** and required by **DevEx** as a Product using the **Theory of Constraints**

Notional View of Platform Engineering

Platform Product Management

Team Topologies, Technical Product Management,
Value Modeling

Developer Plane

Version Control, Infrastructure as Code , Dev Tools, Paved Road

Compliance & Governance Plane

Pipelines , Lightweight governance, FinOps compliance, Compliance @ POC

Delivery & Runtime Plane

Containers, Kubernetes, Workflow orchestration

Networking & Connectivity Plane

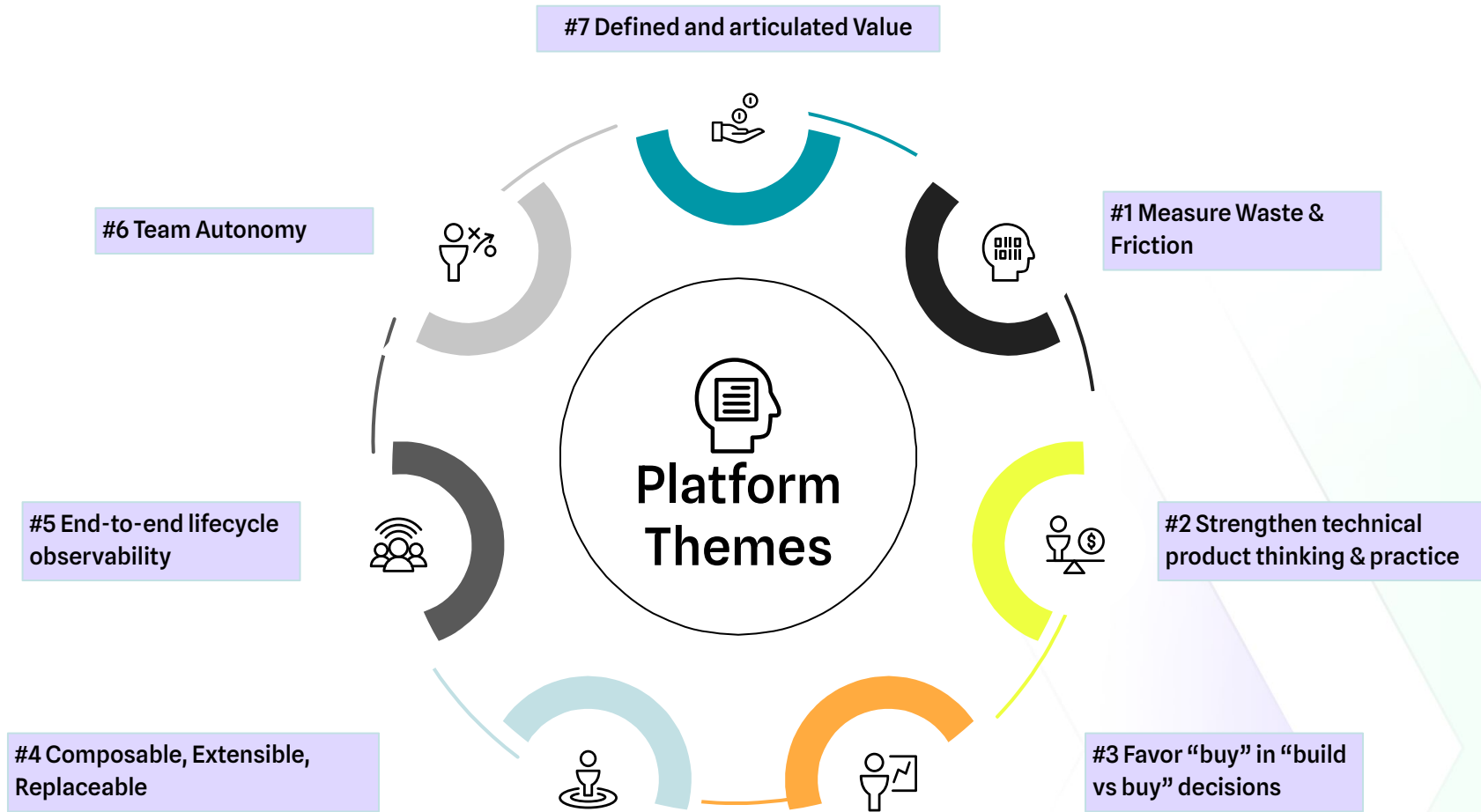
VPC, External, 3rd party

Security Plane

IAM, Secret and Encryption Management, SIEM

Observability

System level, Integrations, Alerting



What are practitioners saying?

PLATFORM ENGINEERING'S KEY BENEFITS FOR DEVELOPERS:



From the state of platform engineering report 2024 – <https://www.puppet.com/resources/state-of-platform-engineering>

Evolving Platform Engineering in Organizations

Why PE Emerged?

Drivers

Complexity of Modern Stacks

Developer Productivity Concerns

Scaling DevOps Practices

Demand for Standardization



Genesis

Self-Identification of Pain Points

Grassroots efforts

Executive Sponsorships

MVP from “Platform” Team



Struggles

Over Engineering

Lack of developer buy-in

Cultural Resistance

Misaligned Priorities

Ideal PE Journey for Organizations

Platform engineering is not just about building tools; it's about building trust. If developers don't see the value in the platform, it doesn't matter how technically brilliant it is



"Start small, deliver value quickly, and iterate. The goal of an MVP is not to be perfect but to learn and improve." – Eric Ries, The Lean Startup

Summary of PE evolution Patterns

Pattern	Who?	How?	Evolution	Example
Grassroots	Devs/Infra who are frustrated with inefficiencies	Build ad-hoc tools or automation to solve specific pain points	Organic adoption → Leadership recognition	Self-service K8s cluster provisioning tool to reduce setup time
Management Driven	Engineering Leadership	Identifies inefficiencies and scalability challenges	Allocate resources → Mandates to solve specific problems	IDP to standardize deployments
Hybrid	1 & 2	Leadership notes grassroots efforts	Informal → Formal	CI/CD Pipeline automation becoming mainstream
Crisis-Driven	Business & Engineering Leadership	Reactive response to a business situation	Major Incidents → Fast Track	Frequent production outages due to inconsistent deployment processes
Vendor Driven	Unicorn vendor products	Industry standard tools become difficult to use and manage	IDP solutions → Platform abstractions	K8s users builds an IDP to simplify cluster management and provide developers with self-service capabilities
Scaling DevOps	Existing DevOps Teams	Natural evolution of modern DevOps practices	Automation → Reusability → Self-Serve	Multiple DevOps teams builds a platform to standardize CI/CD pipelines and infrastructure provisioning

METFLIX

Spotify

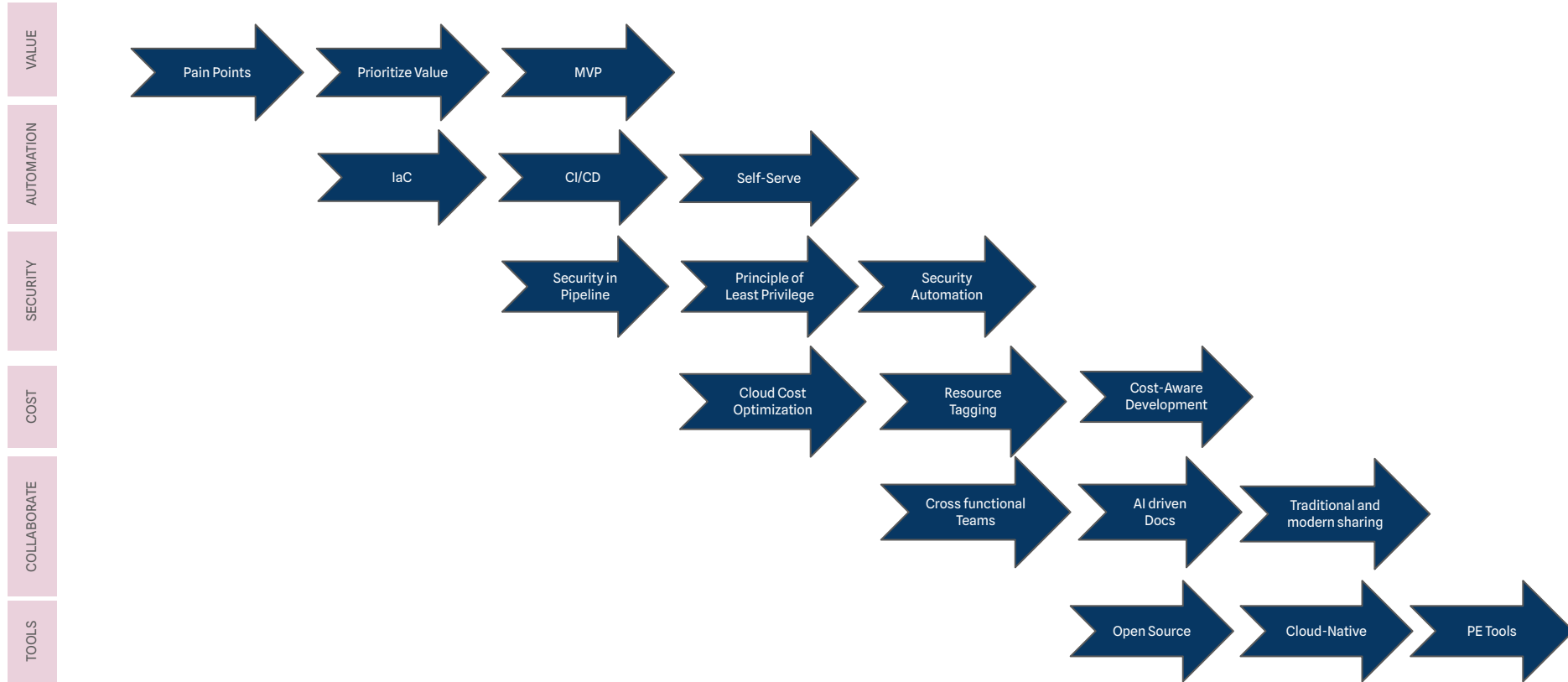
airbnb

Uber

shopify

TARGET

Approach for Midsize Organizations



Gather Feedback | Measure Metrics | Continuous Improvement

Approach for Hybrid Organizations

Challenges

Connectivity and Integrations

Heterogeneous Environments

Security Complexity

Data Residency/Compliance

Legacy Systems



Strategies

Phased Approach

Hybrid Tooling (CI/CD, IaC)

API-First Approach

Abstraction Layer

Security Connectivity

Centralized IAM

Observability Strategy

Data Synchronization



Success Factors

Executive Sponsorship

Cross-Functional Collaboration

Clear Communication

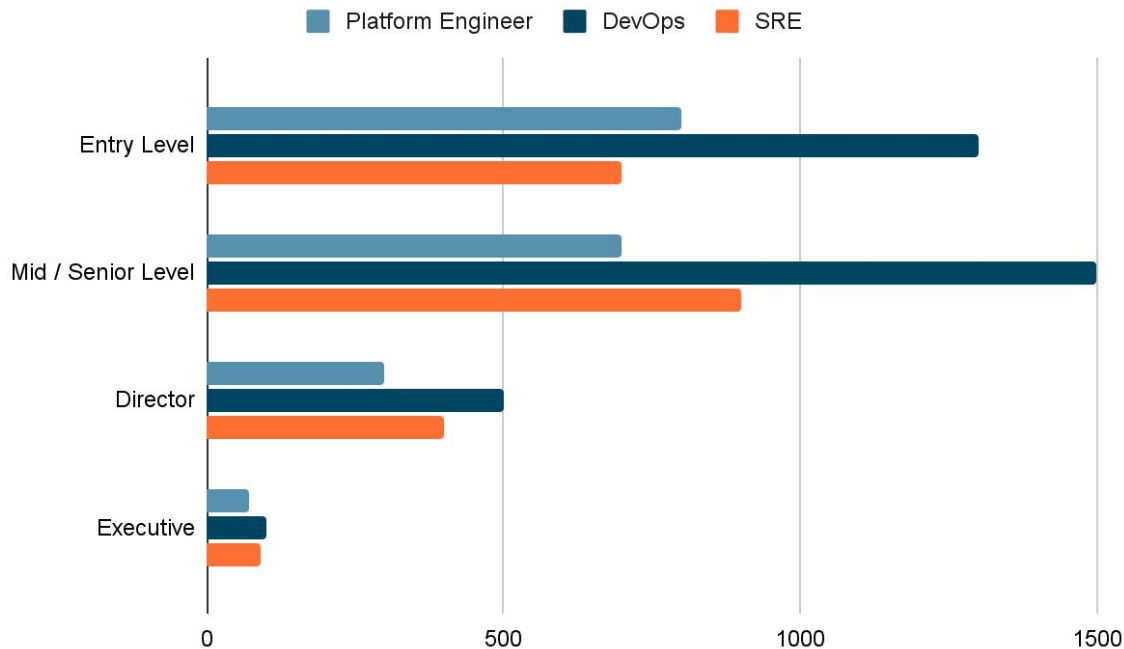
Iterative Approach

Clear Metrics

Platform Engineering Careers

PE Career Progression

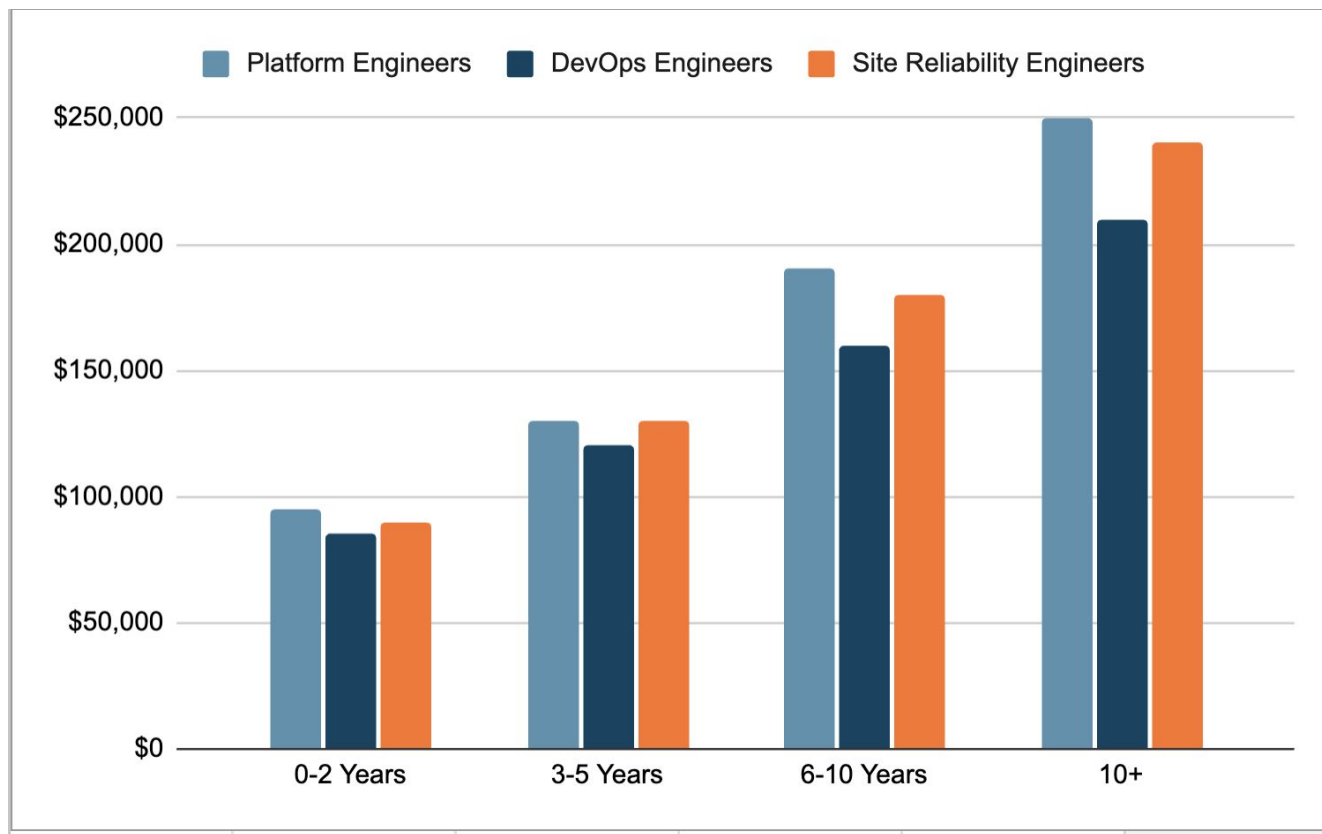
Unscientific search
Across LinkedIn /
Glassdoor/Ziprecruiter



Skill Sets comparison

Skill	PE	DevOps	SRE
Cloud Infra	Cloud Architecture, Build container orchestration	Use platform tools and native services	Use platform tools and native services
Automation CI/CD	IaC, Paved Paths	Use automation to build pipelines, Run it	Self-healing and observability
Programming	Python, Go, Rust, Java, Typescript, JS	Shell, Typescript, Python, Ruby, Groovy	Go, Python
Monitoring/ Observability	Build platforms and capabilities using tools	Configure, report and use tools	Configure and use tools
DevSecOps	Build capabilities	Use capabilities	Validate and report

Salary Progression (Usual Disclaimers Apply)

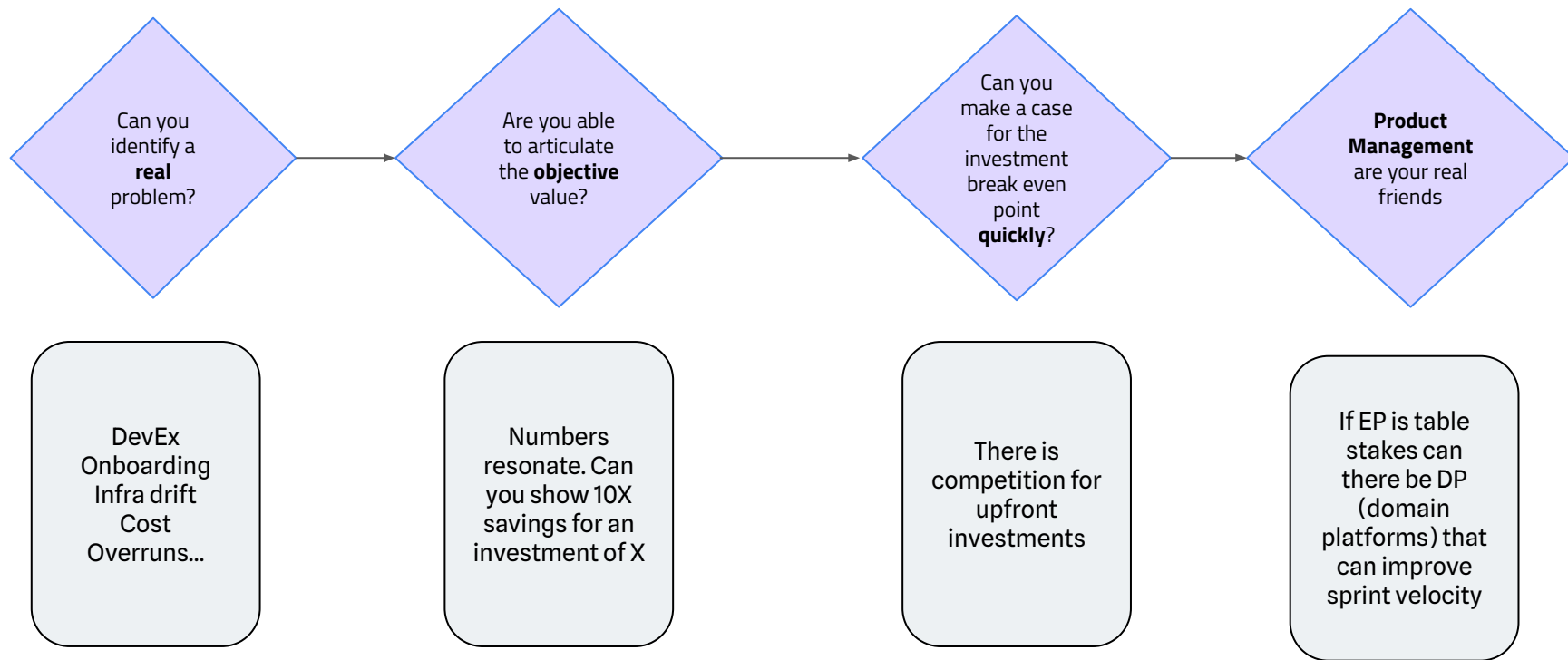


Evolving Skill sets

2010	Virtualization, Basic scripting (Bash, Python), Jenkins, Basic networking, Source control (Git, SVN), Monolithic architecture management
2015	Cloud services (AWS, Azure, GCP), Docker, Configuration management (Chef, Ansible, Puppet), CI/CD, Microservices architecture
2020	Infrastructure as Code (Terraform, CloudFormation), Kubernetes, Serverless architectures, Cloud-native technologies, DevOps methodologies
2025	Agentic AI, AI/ML integration in operations, RAGs, Advanced monitoring and observability, Enhanced security practices, K8s Operators, Multi-cloud management, Progressive delivery
2030	Quantum computing awareness, Advanced AI-driven operations, Sustainability in computing, Edge computing management, Blockchain integration for platform tasks

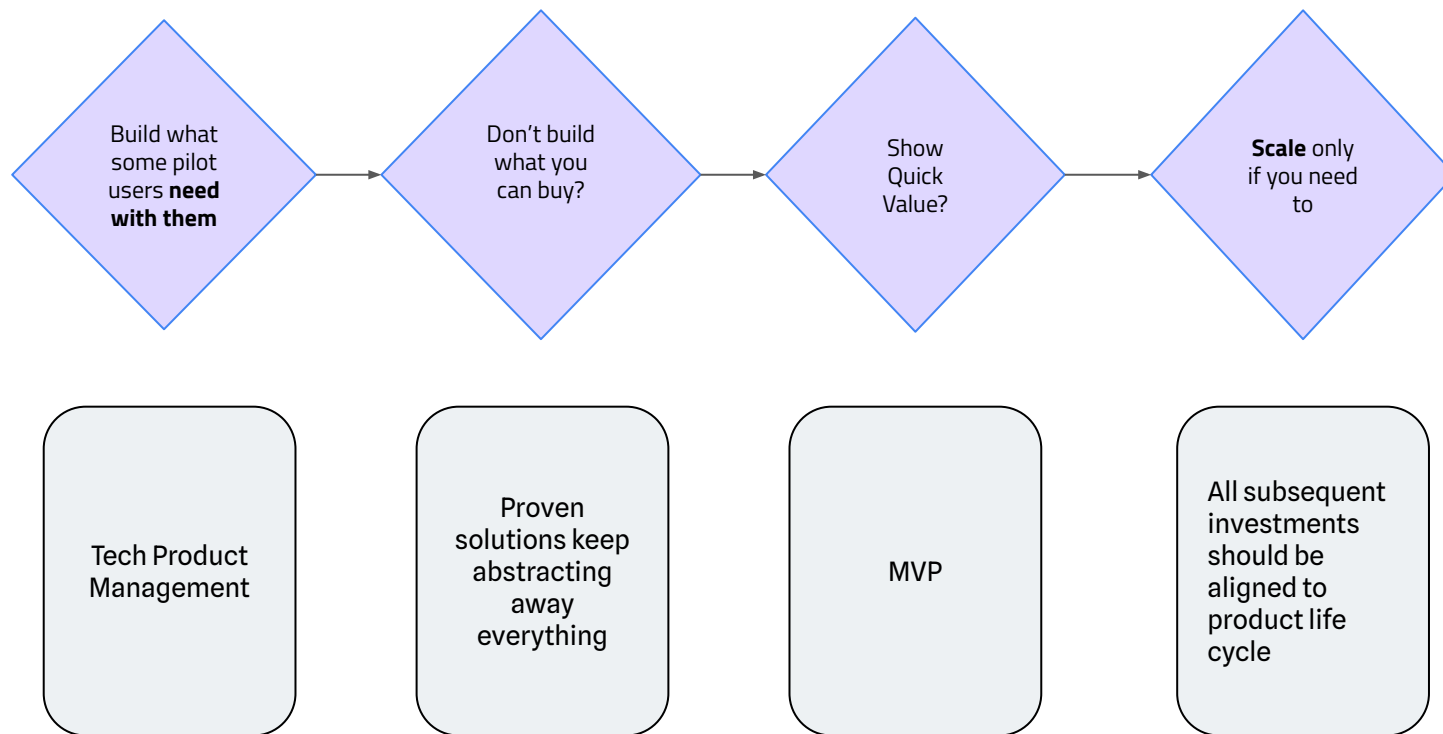
Making Platform Engineering Stick

Selling PE in your organization

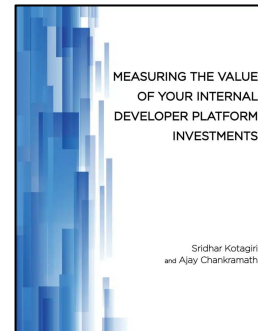


Value Model

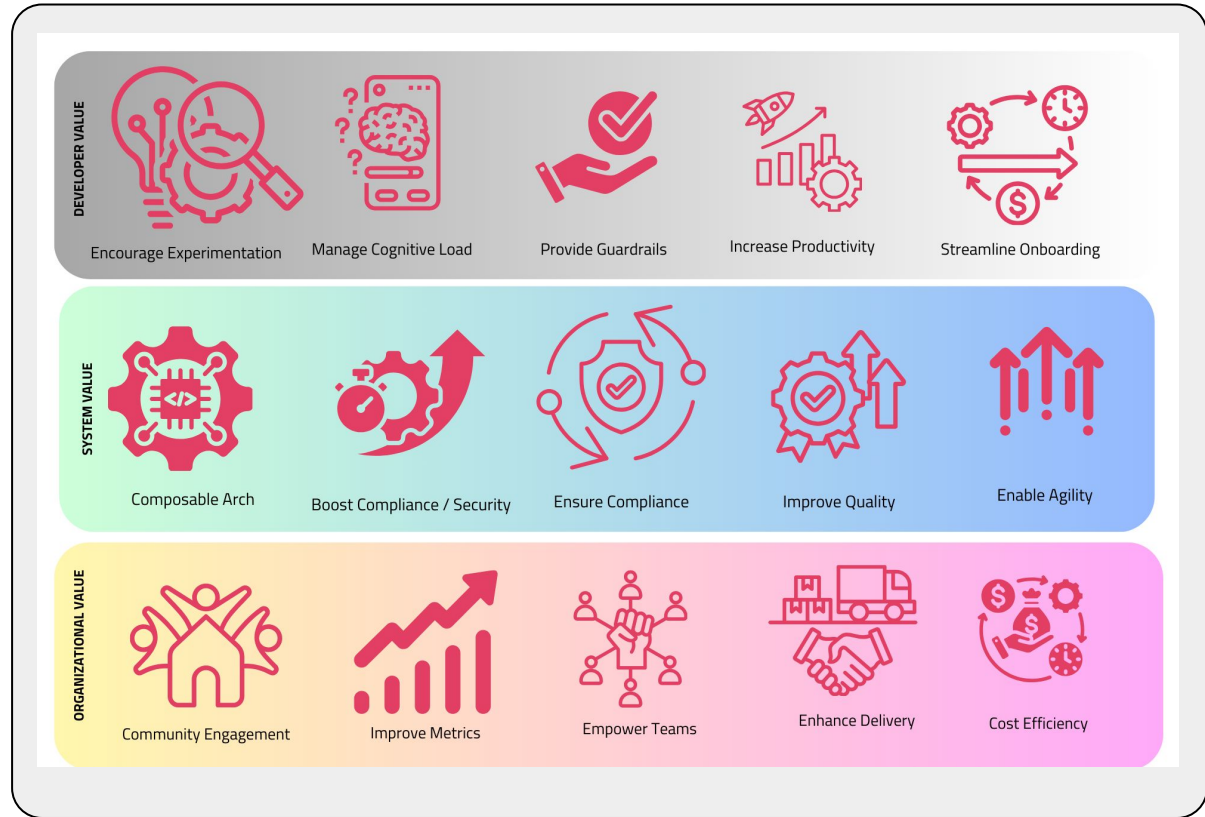
After your sell



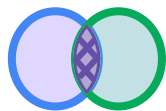
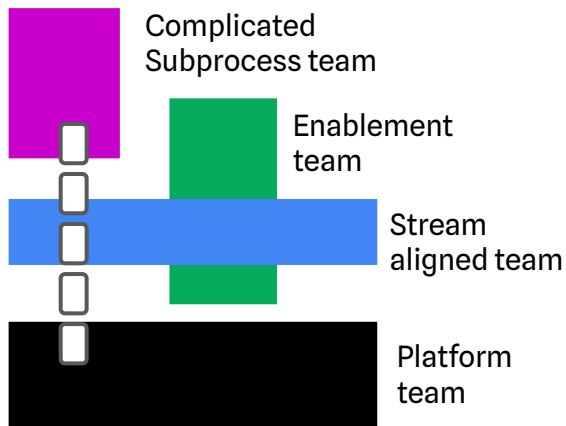
[Value Model](#)



Adoption Conundrum - The Value Proposition

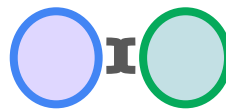


Where do platforms fit in the organization?



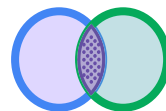
Collaboration

Two teams work together on a shared goal, particularly during discovery of new technology or approaches



X-as-a-Service

One team consumes something provided by another team (such as an API, a tool, or a full software product)

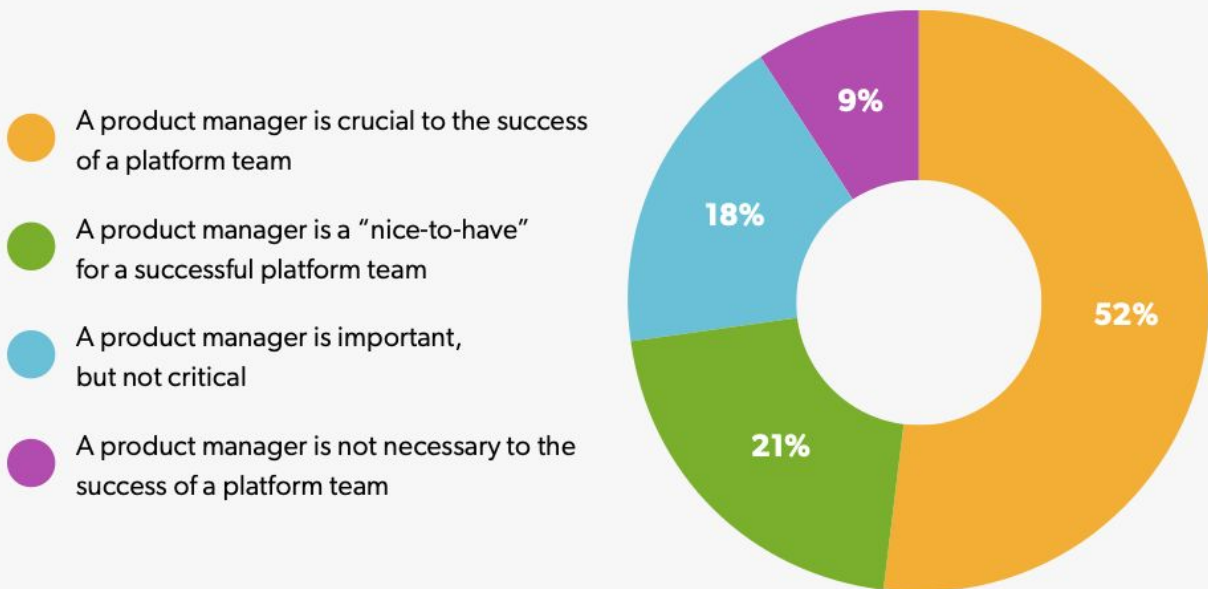


Facilitating

One team (usually an enabling team) facilitates another team in learning or adopting a new approach.

Adoption Conundrum - TPM

HOW IMPORTANT IS A PRODUCT MANAGER TO THE SUCCESS OF A PLATFORM TEAM?



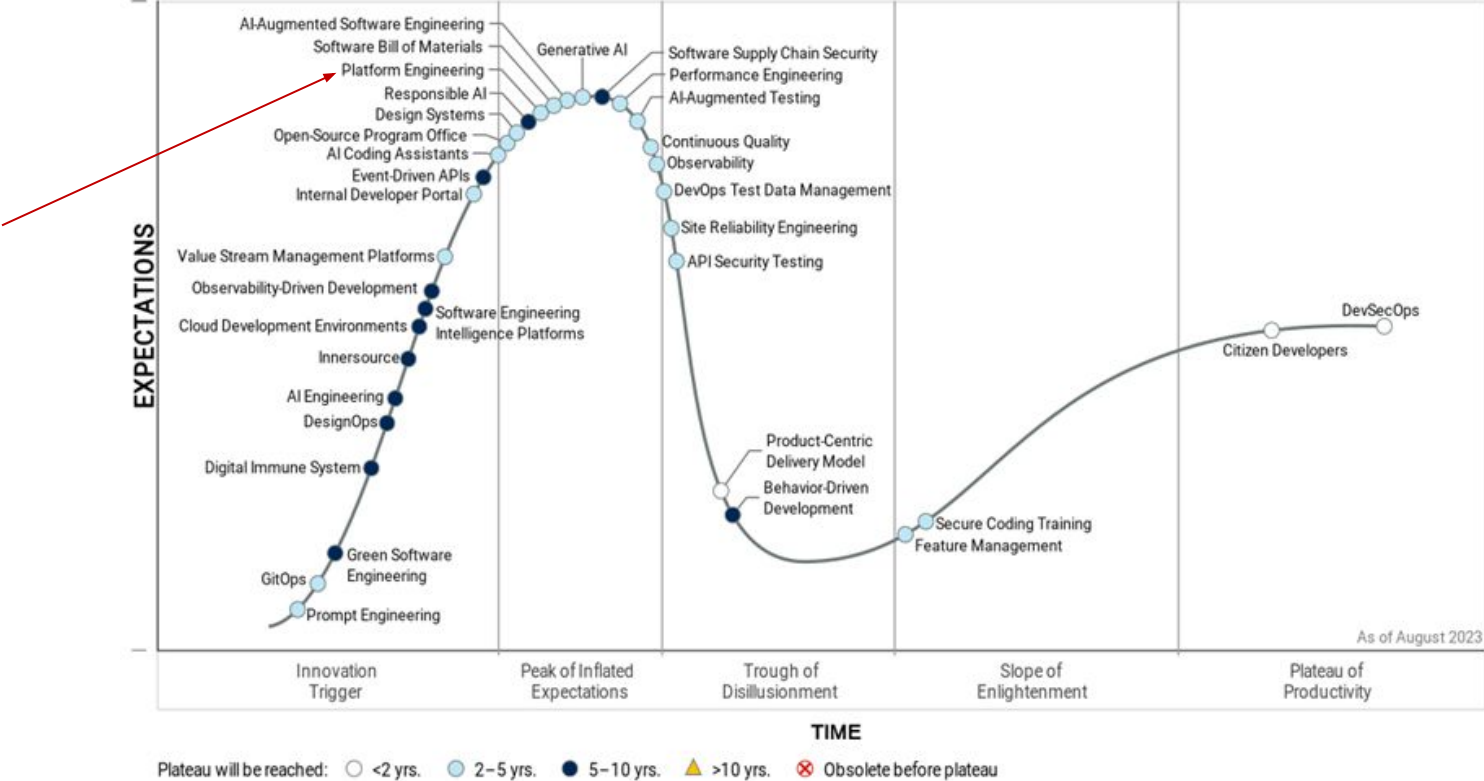
Culture & Organization Matters

“Culture Eats Strategy for Breakfast”

- Attributed to Peter Drucker

What's next in Platform Engineering

Trough of Disillusionment



10 Predictions for the next 10 years



Generative AI

RAGs focused on domain specific activities

Embedded Platform Engineering?

Kubernetes Operators
(Operator Patterns → Advanced Operators)

Platform Convergence?

Experience Platforms, Business Platforms, Engineering Platforms

IDP

Internal Developer Portals → Internal Developer Platforms

GreenOps by default in PE

Too much carbon emissions is no longer an option. Every PE capability will be Sustainability compliant

10 Predictions for the next 10 years



Increase in serverless computing

Led by CSPs providing more Platform Engineering out of the box. Moving abstraction layer?



Advanced Orchestration

Vanishing kubernetes where the orchestration becomes intelligent enough to manage containers through GenAI



Edge Computing

Deployment in edge locations make platform engineering more embedded in products



Agentic Platform Engineering

Combine the principles of platform engineering with the power of agentic AI creating IDPs that are not only self-service but also intelligent and autonomous, capable of anticipating developer needs



Decentralized Operations

Decentralized platforms that ensure data integrity, moving PE beyond Engineering platforms

What you will learn from this book?

10 Things You Will Learn from the book

Understand what Platform Engineering is and why it matters.

Key Domains in a Platform Mental Model

Platform As A Product

Real World Case Studies

Practice, not just theory

Focus on Developer Experience

Team Topologies and why it matters

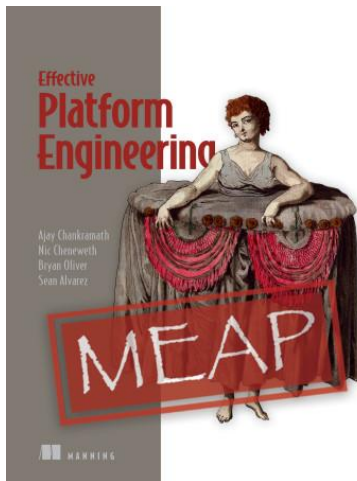
Observability as the backbone

Proven Frameworks

Future Proof with GenAI


Effective Platform Engineering

- **Effective Platform Engineering** book by [Manning](#)
- MEAP currently out awaiting print version soon







More Information



Effective Platform Engineering





[Buy Now](#)


[Ajay Chankramath](#)



[Nic Cheneweth](#)



[Bryan Oliver](#)



[Sean Alvarez](#)


"Effective Platform Engineering" is a comprehensive guide that introduces platform engineering as a discipline, focusing on creating developer platforms that enhance team efficiency and streamline application deployment. The book provides practical insights into designing and managing platforms that bridge the gap between operations and development, automating tasks throughout the software development lifecycle. Readers will learn to build internal developer platforms and portals, ensuring seamless adoption and satisfaction among teams. The authors emphasize the importance of secure, scalable Kubernetes-based engineering platforms and offer strategies for implementing effective Service Level Objectives to boost trust and adoption. Additionally, the book explores cutting-edge integrations of Generative AI tools to enhance developer productivity, providing readers with the knowledge to leverage the latest advancements in code generation.

Through practical examples and real-world scenarios, "Effective Platform Engineering" demonstrates how platform engineering differs from traditional DevOps and the unique value it brings to organizations. The book delves into both patterns and anti-patterns of platform development, guiding readers in designing and deploying secure, scalable, and observable engineering platforms. With the inclusion of diagrams, code samples, and exercises, readers can visualize key concepts and solidify their understanding. This resource is tailored for DevOps engineers familiar with Kubernetes, cloud environments, and infrastructure-as-code, aiming to equip them with the skills to establish platforms that reduce workloads, improve consistency, and accelerate software delivery.

Discover how platform engineering is revolutionizing the developer experience and operational efficiency. [Learn more](#)